TITLE:

A Polynomial Time Sampling Algorithm for an Optimal Family of Min-Wise Independent Permutations (Models of Computation and Algorithms)

AUTHOR(S):

Shinozaki, Takahiro; Itoh, Toshiya

CITATION:

Shinozaki, Takahiro ...[et al]. A Polynomial Time Sampling Algorithm for an Optimal Family of Min-Wise Independent Permutations (Models of Computation and Algorithms). 数理解析研究所講究録 1999, 1093: 74-80

ISSUE DATE:

1999-04

URL:

http://hdl.handle.net/2433/62969

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録
1093 巻 1999 年 74-80

A Polynomial Time Sampling Algorithm for an Optimal Family of Min-Wise Independent Permutations

TAKAHIRO SHINOZAKI staka@ip.titech.ac.jp
TOSHIYA ITOH titoh@ip.titech.ac.jp

Department of Information Processing
Interdisciplinary Graduate School of Science and Engineering
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

1 Introduction

The Web [4] has grown exponentially for recent years and we have a huge amount of identical (or almost identical) documents on the Web at present. When users wish to detect or filter those (almost) identical documents by the currently used indexing software [7], e.g., AltaVista, this might cause them several undesirable problems — (1) indexing of those replicated documents wastes expensive resources such as time and bandwidth; (2) most of users are not interested in getting (almost) identical documents to their queries, etc.

To evaluate the performance of the indexing software, we need to formally define the notion of "almost identical" between documents, however, any of the standard distances defined over strings (e.g., Hamming distance [6] or Levenstien distance [1]) does not capture well our intuition for this notion. In addition, these distances require the pairwise comparison of whole documents. Thus in practice, this is infeasible for a large collection of documents and some sort of sampling for each document is necessary. Now our problem can be reduced to a set intersection problem by *shingling* process [5, 2]. In the shingling process, we associate a set S_D with each document D. In general, we can regard S_D as a set of natural numbers, i.e., there exists a natural number n ($n \simeq 2^{64}$ in practical use) such that $S_D \subseteq [n]$ for each document D, where $[n] = \{1, 2, \ldots, n\}$. Then we can define a good measure (called *resemblance* [2]) to capture the notion of "almost identical" between documents. The resemblance $r(A, B)$ of two documents A and B is defined as $r(A, B) = \|S_A \cap S_B\| / \|S_A \cup S_B\|$, where $\|A\|$ is the cardinality of a finite set A. By experiments, the resemblance is known to capture well the notion of "almost identical." To compute $r(A, B)$, it suffices to consider the sets $S_A, S_B \subseteq [n]$ associated with A, B, respectively, and to evaluate $\Pr(\min\{\pi(S_A)\} = \min\{\pi(S_B)\})$ when π is chosen uniformly at random from S_n, where S_n is the set of all permutations on $[n]$. Indeed, it is easy to show that

$$\Pr(\min\{\pi(S_A)\} = \min\{\pi(S_B)\}) = \frac{\|S_A \cap S_B\|}{\|S_A \cup S_B\|} = r(A, B). \tag{1}$$

Thus the resemblance of two documents can be easily estimated by keeping a *sketch* [2, 3] of each document D. The sketch S_D of D can be efficiently computed and is given by the fixed size list $\mathsf{S}_D = (\min\{\pi_1(S_D)\}, \min\{\pi_2(S_D)\}, \ldots, \min\{\pi_\ell(S_D)\})$ for ℓ (say $\ell \simeq 100$) independently chosen random permutations $\pi_1, \pi_2, \ldots, \pi_\ell \in S_n$. To estimate $r(A, B)$, we only count how many elements in S_A and S_B are common. In practice, however, it is unrealistic to choose π uniformly at random from S_n, because S_n contains a huge number of permutations. Thus in the practical and theoretical point of view, the goal of the paper is to construct a *polynomial time samplable* (smaller) family of permutations that has the property of equation (1).

Definition 1.1 (min-wise independency [3]): *We say that $\mathcal{C} \subseteq S_n$ is a family of* min-wise independent *permutations if for any (nonempty) $X \subseteq [n]$ and any $x \in X$, $\Pr(\min\{\pi(X)\} = \pi(x)) = \|X\|^{-1}$ when π is chosen uniformly at random from $\mathcal{C}$.*

Broder, Charikar, Frieze, and Mitzenmacher [3] showed that $\mathcal{C} \subseteq S_n$ has the property of equation (1) if and only if it is min-wise independent.

Theorem 1.2 (lower bound [3]): *For any integer $n > 0$, let $\mathcal{C} \subseteq S_n$ be a family of min-wise independent permutations. Then $\|\mathcal{C}\| \geq \mathrm{lcm}(n, n-1, \ldots, 2, 1)$ and hence $\|\mathcal{C}\| \geq e^{n-o(n)}$.*

Theorem 1.3 (upper bound [3]): *For any integer $n > 0$, there exists a polynomial time samplable family of min-wise independent permutations $\mathcal{C} \subseteq S_n$ such that $\|\mathcal{C}\| \leq 4^n$.*

Theorem 1.4 (size-optimal family [8]): *For any integer $n > 0$, there exists a family of min-wise independent permutations $\mathcal{F}_n$ such that $\|\mathcal{F}_n\| = \mathrm{lcm}(n, n-1, \ldots, 2, 1)$.*

In this paper, we shall construct a polynomial time samplable family of permutations $\mathcal{F}_n \subseteq S_n$ and then show that $\mathcal{F}_n$ is min-wise independent and $\|\mathcal{F}_n\| = \mathrm{lcm}(n, n-1, \ldots, 2, 1)$.

2 Min-Wise Independent Permutations

Let $\langle a_1, a_2, \ldots, a_k \rangle$ be a *string*. For each $k \in [n]$, let $\mathcal{H}_{n,k} = \{H \subseteq [n] : \|H\| = k\}$ and $\mathcal{P}_{n,k} = \{\langle x_1, x_2, \ldots, x_k \rangle \in [n]^k : x_i \neq x_j \ (i \neq j)\}$. For each $H \in \mathcal{H}_{n,k}$, let $\mathcal{P}^H_{n,k} = \{\langle x_1, x_2, \ldots, x_k \rangle \in \mathcal{P}_{n,k} : x_i \in H\}$. For any $\pi = \langle x_1, x_2, \ldots, x_k \rangle \in \mathcal{P}_{n,k}$ and any $y \in [n] - \{x_1, x_2, \ldots, x_k\}$, we use $\pi \# y$ to denote the concatenation of π and y, i.e., $\pi \# y = \langle x_1, x_2, \ldots, x_k, y \rangle \in \mathcal{P}_{n,k+1}$. For each $k \in [n]$, let $\alpha_{n,k} = \mathrm{lcm}(n, n-1, \ldots, n-k+1)$ and $\beta_{n,k} = \alpha_{n,k}/\alpha_{n,k-1}$, where $\alpha_{n,0} = 1$. From the definition, it is obvious that $\beta_{n,k}$ is an integer, because

$$\beta_{n,k} = \frac{\alpha_{n,k}}{\alpha_{n,k-1}} = \frac{\mathrm{lcm}(\alpha_{n,k-1}, n-k+1)}{\alpha_{n,k-1}} = \frac{(n-k+1)}{\gcd(\alpha_{n,k-1}, n-k+1)}. \tag{2}$$

For any $H \in \mathcal{H}_{n,k}$, let $\pi = \langle x_1, x_2, \ldots, x_k \rangle \in \mathcal{F}^H_{n,k}$ and $\pi' = \langle x'_1, x'_2, \ldots, x'_k \rangle \in \mathcal{F}^H_{n,k}$. Define $\preceq$ on $\mathcal{F}^H_{n,k}$ to be $\pi \preceq \pi'$ if there exists an $i \in [k]$ such that $x_i < x'_i$ and $x_j = x'_j$ for each $i < j \leq k$ or $x_j = x'_j$ for each $j \in [k]$. It is obvious that $\preceq$ is the *total order*. As for the intuition of our construction for the min-wise independent permutations, see [8].

Input: An Integer $n > 0$.
Output: A Family of Permutations $\mathcal{F}_n (= \mathcal{F}_{n,n})$.
Initial: $\mathcal{F}_{n,0} = \mathcal{F}^{\phi}_{n,0} = \{\langle\,\rangle\}$, where λ denotes the null string.
Stage k: For $k = 0, 1, \ldots, n-1$, iterate the following:

Classifying Procedure: For each $H \in \mathcal{H}_{n,k}$, let $\mathcal{F}^H_{n,k} = \{\langle x_1, x_2, \ldots, x_k \rangle \in \mathcal{F}_{n,k} : x_i \in H\}$ and enumerate every $\pi^H_k \in \mathcal{F}^H_{n,k}$ according to the order $\preceq$, where $f_k(H) = \|\mathcal{F}^H_{n,k}\|$. For each $i \in [m_k]$, where $m_k = \binom{n}{k}$, arrange $\pi^{H_i}_{k,1}, \pi^{H_i}_{k,2}, \ldots, \pi^{H_i}_{k,f_k(H_i)}$ in the the array C_k as follows:

The Array C_k

H_1	H_2	$\cdots\cdots\cdots$	H_{m_k}
$\pi^{H_1}_{k,1}$	$\pi^{H_2}_{k,1}$	$\cdots\cdots\cdots$	$\pi^{H_{m_k}}_{k,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\pi^{H_1}_{k,f_k(H_1)}$	$\pi^{H_2}_{k,f_k(H_2)}$	$\cdots\cdots\cdots$	$\pi^{H_{m_k}}_{k,f_k(H_{m_k})}$

Replicating Procedure: For each $H \in \mathcal{H}_{n,k}$, generate $\beta_{n,k+1}$ copies of $\pi^H_{k,i} \in \mathcal{F}^H_{n,k}$ for each $i \in [f_k(H)]$, and enumerate those $\beta_{n,k+1}$ copies of each $\pi^H_{k,i} \in \mathcal{F}^H_{n,k}$ as follows:

$$\underbrace{\begin{matrix}\tilde{\pi}^H_{k,1} & \cdots & \tilde{\pi}^H_{k,\beta_{n,k+1}} \\ \| & \cdots & \| \\ \pi^H_{k,1} & \cdots & \pi^H_{k,1}\end{matrix}}_{\beta_{n,k+1}\text{ copies}} \quad \underbrace{\begin{matrix}\tilde{\pi}^H_{k,\beta_{n,k+1}+1} & \cdots & \tilde{\pi}^H_{k,2\beta_{n,k+1}} \\ \| & \cdots & \| \\ \pi^H_{k,2} & \cdots & \pi^H_{k,2}\end{matrix}}_{\beta_{n,k+1}\text{ copies}} \quad \cdots \quad \underbrace{\begin{matrix}\tilde{\pi}^H_{k,(f_k(H)-1)\beta_{n,k+1}+1} & \cdots & \tilde{\pi}^H_{k,f_k(H)\beta_{n,k+1}} \\ \| & \cdots & \| \\ \pi^H_{k,f_k(H)} & \cdots & \pi^H_{k,f_k(H)}\end{matrix}}_{\beta_{n,k+1}\text{ copies}}$$

For each $i \in [m_k]$, arrange $\tilde{\pi}^{H_i}_{k,1}, \tilde{\pi}^{H_i}_{k,2}, \ldots, \tilde{\pi}^{H_i}_{k,f_k(H_i)\beta_{n,k+1}}$ in the array R_k as follows:

The Array R_k

H_1	H_2	$\cdots\cdots\cdots$	H_{m_k}
$\tilde{\pi}^{H_1}_{k,1}$	$\tilde{\pi}^{H_2}_{k,1}$	$\cdots\cdots\cdots$	$\tilde{\pi}^{H_{m_k}}_{k,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\tilde{\pi}^{H_1}_{k,f_k(H_1)\beta_{n,k+1}}$	$\tilde{\pi}^{H_2}_{k,f_k(H_2)\beta_{n,k+1}}$	$\cdots\cdots\cdots$	$\tilde{\pi}^{H_{m_k}}_{k,f_k(H_{m_k})\beta_{n,k+1}}$

Appending Procedure: For each $H \in \mathcal{H}_{n,k}$, let $H^c = [n] - H = \{y_1, y_2, \ldots, y_{n-k}\}$ and assume that $y_1 < y_2 < \cdots < y_{n-k}$. Let $\mathcal{G}^H_{n,k+1} = \{\pi^H_{k+1,i} : \pi^H_{k+1,i} = \tilde{\pi}^H_{k,i} \# y_{\mathsf{res}(i-1,n-k)+1}$ for each $i \in [f_k(H)\beta_{n,k+1}]\}$, where $\mathsf{res}(a,b)$ is the function that returns the residue when b divides a. For each $i \in [m_k]$, arrange $\pi^{H_i}_{k+1,1}, \pi^{H_i}_{k+1,2}, \ldots, \pi^{H_i}_{k+1,f_k(H_i)\beta_{n,k+1}}$ in the array A_{k+1} as follows:

The Array A_{k+1}

H_1	H_2	$\cdots\cdots\cdots$	H_{m_k}
$\pi^{H_1}_{k+1,1}$	$\pi^{H_2}_{k+1,1}$	$\cdots\cdots\cdots$	$\pi^{H_{m_k}}_{k+1,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\pi^{H_1}_{k+1,f_k(H_1)\beta_{n,k+1}}$	$\pi^{H_2}_{k+1,f_k(H_2)\beta_{n,k+1}}$	$\cdots\cdots\cdots$	$\pi^{H_{m_k}}_{k+1,f_k(H_{m_k})\beta_{n,k+1}}$

Finally, define $\mathcal{F}_{n,k+1} = \cup_{H \in \mathcal{H}_{n,k}} \mathcal{G}^H_{n,k+1}$, i.e., a collection of all entries in the array A_{k+1}.

For each $\pi \in \mathcal{F}_n$, we regard $\pi = \langle x_1, x_2, \ldots, x_n \rangle$ as a permutation on $[n]$ in the natural manner, i.e., π defines a permutation on $[n]$ in such a way that $\pi(x_i) = i$ for each $i \in [n]$.

For our construction of $\mathcal{F}_n \subseteq S_n$, the following lemma and proposition are essential to guarantee the min-wise independency of $\mathcal{F}_n$ and $\|\mathcal{F}_n\| = \mathrm{lcm}(n, n-1, \ldots, 2, 1)$.

Lemma 2.1 [8]: *For each $0 \le k \le n-1$, the following holds:*

(a) *$\alpha_{n,k}$ is divisible by $\binom{n}{k}$;* (b) *$\alpha_{n,k}\beta_{n,k+1} = \alpha_{n,k+1}$ is divisible by $(n-k)\binom{n}{k}$.*

Proposition 2.2 [8]: *For each $0 \le k \le n-1$ and each $H \in \mathcal{H}_{n,k}$, the following holds:*

(a) $\|\mathcal{F}^H_{n,k}\| = \alpha_{n,k}/\binom{n}{k}$; (b) $\|\mathcal{G}^H_{n,k+1}\| = \alpha_{n,k+1}/\binom{n}{k}$.

By the use of Lemma 2.1 and Proposition 2.2, we can show the following:

Theorem 2.3 [8]: *For any integer $n > 0$, the family of permutations $\mathcal{F}_n$ is min-wise independent and $\|\mathcal{F}_n\| = \alpha_{n,n} = \mathrm{lcm}(n, n-1, \ldots, 2, 1)$.*

3 Polynomial Time Samplability

3.1 Overview of Our Polynomial Time Sampling Algorithm

We first describe the intuition behind our polynomial time sampling algorithm Samp.

For each $H \in \mathcal{H}_{n,k}$ and each $x \in [n]$, let $\mathcal{E}_{k+1}^{(H,x)} = \{\langle x_1, x_2, \ldots, x_k, x_{k+1}\rangle \in \mathcal{G}_{n,k+1}^{H} : x_{k+1} = x\}$. The following lemma plays an essential role to design the sampling algorithm Samp.

Lemma 3.1 [8, Eq.(11)]: $\left\|\mathcal{E}_{k+1}^{(H,x)}\right\| = \alpha_{n,k+1}/\{(k+1)\binom{n}{k+1}\}$ *for each* $H \in \mathcal{H}_{n,k}$ *and each* $x \in [n]$.

Remark 3.2: For each $H = \{x_1, x_2, \ldots, x_{k+1}\} \in \mathcal{H}_{n,k+1}$ and each $x \in H$, we define $\mathcal{T}_{k+1}^{H}(x) = \{\langle \xi_1, \xi_2, \ldots, \xi_{k+1}\rangle \in \mathcal{F}_{n,k+1}^{H} : \xi_{k+1} = x\}$, i.e., $\mathcal{T}_{k+1}^{H}(x)$ is the set of $\pi \in \mathcal{F}_{n,k+1}^{H}$ that has $x \in H$ at the $(k+1)$-th (rightmost) position. Note that $\mathcal{E}_{k+1}^{(H-\{x\},x)} = \mathcal{T}_{k+1}^{H}(x)$. So it follows from Lemma 3.1 that $\mathcal{F}_{n,k+1}^{H}$ can be partitioned into $k+1$ sets $\mathcal{T}_{k+1}^{H}(x_i)$ of equal size.

For each $0 \le k \le n-1$, let $L_k(\pi)$ be the location of $\pi \in \mathcal{F}_{n,k}^{H}$ with respect to the total order $\preceq$ on $\mathcal{F}_{n,k}^{H}$, where $0 \le L_k(\pi) \le \alpha_{n,k}/\binom{n}{k} - 1$ and $L_0(\pi_0) = L_0(\langle\ \rangle) = 0$. In Step 0, the algorithm Samp randomly chooses $x \in [n]$ and sets $\pi_1 = \langle x\rangle$ and $H_1 = \{x\}$. Assume that in Step k, the algorithm Samp keeps $\pi_k = \langle x_1, x_2, \ldots, x_k\rangle \in \mathcal{F}_{n,k}^{H}$, $H_k = \{x_1, x_2, \ldots, x_k\} \in \mathcal{H}_{n,k}$, and $L_k(\pi_k)$. Let $\Gamma_{n,k+1} = (n-k)/\beta_{n,k+1}$. Then from equation (2), we have that

$$\Gamma_{n,k+1} = \frac{n-k}{\beta_{n,k+1}} = (n-k)\cdot\frac{\gcd(\alpha_{n,k}, n-k)}{n-k} = \gcd(\alpha_{n,k}, n-k).$$

Let $H_k^c = [n] - H_k = \{y_1, y_2, \ldots, y_{n-k}\}$ and $y_1 < y_2 < \cdots < y_{n-k}$. Partition H_k^c into $\Gamma_{n,k+1}$ blocks of size $\beta_{n,k+1}$, i.e., $H_{k,i}^c = \{y_{i\cdot\beta_{n,k+1}+1}, y_{i\cdot\beta_{n,k+1}+2}, \ldots, y_{(i+1)\cdot\beta_{n,k+1}}\}$ for each $0 \le i \le \Gamma_{n,k+1} - 1$. Recall our construction of the family $\mathcal{F}_n$. In Stage k, we generate $\beta_{n,k+1}$ copies of each entry in the column index by $H_k \in \mathcal{H}_{n,k}$ in the Replicating Procedure and cyclically append $y_j \in H_k^c$ in the Appending Procedure. Then we can append any $y \in H_{k,\mathsf{res}(L_k(\pi_k),\Gamma_{n,k+1})}^{c}$ to $\pi_k \in \mathcal{F}_{n,k}^{H_k}$. We call $H_{k,\mathsf{res}(L_k(\pi_k),\Gamma_{n,k+1})}^{c}$ an *allowable* block for $\pi \in \mathcal{F}_{n,k}^{H_k}$ at the Appending Procedure in Stage k. Thus the sampling algorithm Samp randomly chooses $y \in H_{k,\mathsf{res}(L_k(\pi_k),\Gamma_{n,k+1})}^{c}$ and sets $H_{k+1} = H_k \cup \{x_{k+1}\} \in \mathcal{H}_{n,k+1}$ and $\pi_{k+1} = \pi_k \# x_{k+1} \in \mathcal{F}_{n,k+1}^{H_{k+1}}$, where $x_{k+1} = y$.

As a next task, Samp needs to evaluate $L_{k+1}(\pi_{k+1})$ (see Figure 1). Let $\Delta_{k+1}(x_{k+1})$ be the location of x_{k+1} in the set $H_{k+1} = \{x_1, x_2, \ldots, x_{k+1}\}$, where $0 \le \Delta_{k+1}(x_{k+1}) \le k$, when $x_i \in H_{k+1}$ are arranged in the increasing order, and let $\delta_{k+1}(\pi_{k+1})$ be the location of $\pi_{k+1} \in \mathcal{T}_{k+1}^{H_{k+1}}(x_{k+1})$ with respect to the total order $\preceq$ on $\mathcal{T}_{k+1}^{H_{k+1}}(x_{k+1})$, where $0 \le \delta_{k+1}(\pi_{k+1}) \le \alpha_{n,k+1}/\{(k+1)\binom{n}{k+1}\} - 1$. We note that $\pi_{k+1} \in \mathcal{F}_{n,k+1}^{H_{k+1}}$ and each entry in $\mathcal{F}_{n,k+1}^{H_{k+1}}$ is arranged according to the total order $\preceq$ at the Classifying Procedure in Stage $k+1$. Then it follows from Remark 3.2 that

$$L_{k+1}(\pi_{k+1}) = \frac{\alpha_{n,k+1}}{(k+1)\binom{n}{k+1}}\cdot\Delta_{k+1}(x_{k+1}) + \delta_{k+1}(\pi_{k+1}).$$

Note that every $\pi \in \mathcal{T}_{k+1}^{H_{k+1}}(x_{k+1})$ has x_{k+1} as the rightmost element. Thus $\pi \in \mathcal{T}_{k+1}^{H_{k+1}}(x_{k+1})$ iff there exists $\pi' \in \mathcal{F}_{n,k}^{H_k}$ to which x_{k+1} is appended at the Appending Procedure in Stage k. Since π_k is located at the $L_k(\pi_k)$-th position in $\mathcal{F}_{n,k}^{H_k}$, it suffices for the evaluation of $\delta_{k+1}(\pi_{k+1})$ to count

the number of $\pi \in \mathcal{F}_{n,k}^{H_k}$ such that $\mathsf{res}(L_k(\pi), \Gamma_{n,k+1}) = \mathsf{res}(L_k(\pi_k), \Gamma_{n,k+1})$ and $L_k(\pi) < L_k(\pi_k)$. So $\delta_{k+1}(\pi_{k+1}) = \lfloor L_k(\pi_k)/\Gamma_{n,k+1} \rfloor$ and we finally have that

$$L_{k+1}(\pi_{k+1}) = \frac{\alpha_{n,k+1}}{(k+1)\binom{n}{k+1}} \cdot \Delta_{k+1}(x_{k+1}) + \left\lfloor \frac{L_k(\pi_k)}{\Gamma_{n,k+1}} \right\rfloor .$$

It is obvious that by the use of $L_{k+1}(\pi_{k+1})$, the algorithm Samp can specify the allowable block $H_{k+1,i}^c$ for $\pi_{k+1} \in \mathcal{F}_{n,k+1}^{H_{k+1}}$ at the Appending Procedure in Stage $k+1$.

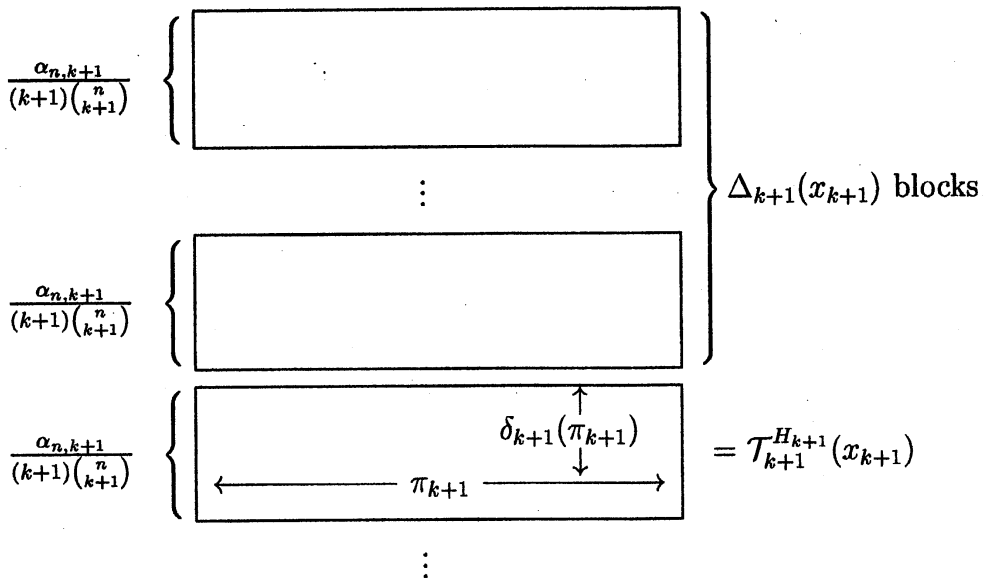

Figure 1: Evaluation of $L_{k+1}(\pi_{k+1})$

3.2 Description of the Algorithm

The following is the formal description of the sampling algorithm Samp for the optimal family of min-wise independent permutations.

Input: An integer $n > 0$.
Output: $\pi \in \mathcal{F}_n$.
Initial: Let $\pi_0 = \langle\ \rangle$, $H_0 = \phi$, and $L_0(\pi_0) = L_0(\langle\ \rangle) = 0$.
Step k: For $k = 0, 1, \ldots, n-1$, iterate the following:

(1) Let $H_k^c = [n] - H_k = \{y_1, y_2, \ldots, y_{n-k}\}$, where $y_1 < y_2 < \cdots < y_{n-k}$, and partition H_k^c into $\Gamma_{n,k+1} = (n-k)/\beta_{n,k+1}$ blocks of size $\beta_{n,k+1}$, i.e., for each $0 \le i \le \Gamma_{n,k+1} - 1$, let $H_{k,i}^c = \{y_{i \cdot \beta_{n,k+1}+1}, y_{i \cdot \beta_{n,k+1}+2}, \ldots, y_{(i+1) \cdot \beta_{n,k+1}}\}$.

(2) Compute $\mathsf{res}(L_k(\pi_k), \Gamma_{n,k+1})$ and choose $y \in H_{k,\mathsf{res}(L_k(\pi_k),\Gamma_{n,k+1})}^c$ uniformly at random.

(3) Let $x_{k+1} = y$ and set $H_{k+1} = H_k \cup \{x_{k+1}\} \in \mathcal{H}_{n,k+1}$ and $\pi_{k+1} = \pi_k \# x_{k+1} \in \mathcal{F}_{n,k+1}^{H_{k+1}}$.

(4) Let $0 \le \Delta_{k+1}(x_{k+1}) \le k$ be the location of x_{k+1} in $H_{k+1} = \{x_1, x_2, \ldots, x_{k+1}\}$ when we arrange $x_i \in H_{k+1}$ in the increasing order, and evaluate

$$L_{k+1}(\pi_{k+1}) = \frac{\alpha_{n,k+1}}{(k+1)\binom{n}{k+1}} \cdot \Delta_{k+1}(x_{k+1}) + \left\lfloor \frac{L_k(\pi_k)}{\Gamma_{n,k+1}} \right\rfloor .$$

3.3 Analysis of the Algorithm

In this subsection, we analyze the time complexity of the sampling algorithm Samp and show that the algorithm Samp runs in polynomial time in n (not in the length of n).

To analyze the time complexity of the sampling algorithm Samp, we need to estimate the time complexity of integer multiplication and integer division. For integers $m \geq \ell > 0$, we use $\mathrm{MULT}(m, \ell)$ to denote the number of bit operations required to multiply an m bit integer by an ℓ bit integer, and $\mathrm{QUOT}(m, \ell)$ and $\mathrm{RES}(m, \ell)$ to denote the number of bit operations required to compute the quotient and residue, respectively, when dividing an m bit integer by an ℓ bit integer. We also use $\mathrm{EUC}(m, \ell)$ to denote the number of bit operations required to compute the greatest common divisor (gcd) of m bit integer and an ℓ bit integer.

Lemma 3.3: *For any integer* $m \geq \ell > 0$, $\mathrm{MULT}(m, \ell) = O(m \lg m \lg\lg m)$.

Lemma 3.4: *For any integer* $m \geq \ell > 0$, $\mathrm{QUOT}(m, \ell) = O(m\ell)$ *and* $\mathrm{RES}(m, \ell) = O(m\ell)$. *If* $0 < m \leq 2\ell$, *then* $\mathrm{QUOT}(m, \ell) = O(\ell \lg \ell \lg\lg \ell)$ *and* $\mathrm{RES}(m, \ell) = O(\ell \lg \ell \lg\lg \ell)$.

Lemma 3.5: *For any integer* $m \geq \ell > 0$, $\mathrm{EUC}(m, \ell) = O(m\ell + \ell \lg^2 \ell \lg\lg \ell)$. *If* $0 < m \leq 2\ell$, *then* $\mathrm{EUC}(m, \ell) = O(\ell \lg^2 \ell \lg\lg \ell)$.

To specify the allowable block $H^c_{k+1,i}$ for $\pi_k \in \mathcal{F}^{H_k}_{n,k}$ and the location $L_{k+1}(\pi_{k+1})$ of π_{k+1} in $\mathcal{F}^{H_{k+1}}_{n,k+1}$ in Step k, the algorithm Samp evaluates: (1) $\Gamma_{n,k+1}$; (2) $\mathsf{res}(L_k(\pi_k), \Gamma_{n,k+1})$; (3) $\alpha_{n,k+1}$; (4) $A_{n,k+1} = \alpha_{n,k+1}/\{(k+1)\binom{n}{k+1}\}$; and (5) $\delta_{k+1}(\pi_{k+1}) = \lfloor L_k(\pi_k)/\Gamma_{n,k+1} \rfloor$. By definitions, we can immediately derive the following proposition:

Proposition 3.6: *For each* $0 \leq k \leq n-1$, *the following holds*: (1) $\Gamma_{n,k+1} = \gcd(\alpha_{n,k}, n-k)$; (2) $\alpha_{n,k+1} = \{\alpha_{n,k} \cdot (n-k)\}/\Gamma_{n,k+1}$; *and* (3) $A_{n,k+1} = (A_{n,k} \cdot k)/\Gamma_{n,k+1}$.

Since $\alpha_{n,1} = n$, $A_{n,1} = 1$, and $L_1(\pi_1) = 0$ in Step 0, assume that in Step k, the algorithm Samp already has $\alpha_{n,k}$, $A_{n,k}$, and $L_k(\pi_k)$ and recursively evaluates $\alpha_{n,k+1}$, $A_{n,k+1}$, and $L_{k+1}(\pi_{k+1})$.

In general, $\alpha_{n,k}$, $A_{n,k}$, and $L_k(\pi_k)$ are n bit integers, and $n-k$, $\Gamma_{n,k+1}$, and k are $\lg n$ bit integers in Step k. Then the following holds for the time complexity of Samp in Step k.

(1) Time complexity of evaluating $\mathsf{res}(L_k(\pi_k), \Gamma_{n,k+1})$: $\mathrm{RES}(n, \lg n) = O(n \lg n)$ bit operations are required (from Lemma 3.4).

(2) Time complexity of evaluating $\Gamma_{n,k+1}$: $\mathrm{EUC}(n, \lg n) = O(n \lg n)$ bit operations are required (from Proposition 3.6-(1) and Lemma 3.5).

(3) Time complexity of evaluating $\alpha_{n,k+1}$: $\mathrm{QUOT}(\lg n, \lg n) = O(\lg n \lg\lg n \lg\lg\lg n)$ and $\mathrm{MULT}(n, \lg n) = O(n \lg n \lg\lg n)$ bit operations are required (from Proposition 3.6-(2) and Lemmas 3.4 and 3.3), i.e., $O(n \lg n \lg\lg n)$ bit operations are required.

(4) Time complexity of evaluating $A_{n,k+1} = \alpha_{n,k+1}/\{(k+1)\binom{n}{k+1}\}$: $\mathrm{MULT}(n, \lg n) = O(n \lg n \lg\lg n)$ and $\mathrm{QUOT}(n, \lg n) = O(n \lg n)$ bit operations are required (from Proposition 3.6-(3) and Lemmas 3.4 and 3.3), i.e., $O(n \lg n \lg\lg n)$ bit operations are required.

(5) Time complexity of evaluating $\delta_{k+1}(\pi_{k+1}) = \lfloor L_k(\pi_k)/\Gamma_{n,k+1} \rfloor$: $\mathrm{QUOT}(n, \lg n) = O(n \lg n)$ bit operations are required (from Lemma 3.4).

Thus in Step k of the algorithm Samp, $O(n \lg n \lg\lg n)$ bit operations are required. Since the algorithm Samp iterates this process n times, we finally have the following theorem:

Theorem 3.7: *For any* $n > 0$, *the algorithm* Samp *requires* $O(n^2 \lg n \lg\lg n)$ *bit operations to uniformly sample* π *from the (smallest) family of min-wise independent permutations* $\mathcal{F}_n$.

4 Concluding Remarks

In this paper, we have constructed an optimal family of min-wise independent permutations $\mathcal{F}_n$ in the sense of family size, and have shown that it is polynomial time samplable.

Let $k \in [n]$ be a fixed integer ($k \simeq 1000$ in general). In the practical point of view, Broder, et.al. [3] defined the notion of "k-restricted min-wise independent."

Definition 4.1 (restricted min-wise independency [3]): *We say that $\mathcal{C} \subseteq S_n$ is a family of k-restricted min-wise independent permutation if for any $X \subseteq [n]$ such that $1 \leq \|X\| \leq k \leq n$ and any $x \in X$, $\Pr(\min\{\pi(X)\} = \pi(x)) = \|X\|^{-1}$ when π is chosen uniformly at random from $\mathcal{C}$.*

In a way similar to Theorem 1.2, Broder, et.al. [3] derived the lower bound below for any family of k-restricted min-wise independent permutations.

Theorem 4.2 [3]: *For any integer $n > 0$, let $\mathcal{C} \subseteq S_n$ be a family of k-restricted min-wise independent permutations. Then $\|\mathcal{C}\| \geq \mathrm{lcm}(k, k-1, \ldots, 2, 1)$ and hence $\|\mathcal{C}\| \geq e^{k-o(k)}$.*

At present, we do not know whether for any integers $n > 0$ and $0 < k < n$, there exists a family of k-restricted min-wise (but not min-wise) independent permutations, but our construction of the family of min-wise independent permutations could be applied to this.

References

[1] Aho, A.V., "Algorithms for Finding Patterns in Strings," in Handbook of Theoretical Computer Science, Vol.A, MIT Press/Elsevier (1990).

[2] Broder, A.Z., On the Resemblance and Containment of Documents," *Proc. of Compression and Complexity of Sequences*, pp.21-29 (1998).

[3] Broder, A.Z., Charikar, M., Frieze, A.M., and Mitzenmacher, M., "Min-Wise Independent Permutations," *Proc. of the Thirtieth Annual ACM Symposium on Theory of Computing*, pp.327-336 (1998).

[4] Berners-Lee, T., Cailliau, R., Loutonen, A., Nielsen, H.F., and Secret, A., "The World-Wide Web," *Communications of the ACM*, Vol.37, No.1, pp.76-82 (1994).

[5] Broder, A.Z., Glassman, S.C., Manasse, M.S., and Zweig, G., "Syntactic Clustering of the Web," *Proc. of the Sixth International World Wide Web Conference*, pp.391-404 (1997).

[6] MacWilliams, F.J. and Sloane, N.J.A., "The Theory of Error-Correcting Codes," 9th edition, North-Holland (1996).

[7] Souza, R.J., Krishnakumar, P., Özveren, C.M., Simcoe, R.J., Spinney, B.A., Thomas, R.E., and Walsh, R.J., "GIGAswitch: A High-Performance Packet-Switching Platform," *DIGITAL Technical Journal*, Vol.6, No.1, pp.9-22 (1994).

[8] Takei, Y., Itoh, T., and Shinozaki, T., "An Optimal Construction of Exactly Min-Wise Independent Permutations," *Technical Report of IEICE*, COMP98-62, pp.89-98 (1998).